(12)

AD A111300

DTIC
SELECTE
FEB 23 1982
B

DTIC FILE COPY

8 2 02 11 005

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

(12)

THE SORTIE-GENERATION MODEL SYSTEM
VOLUME III
SORTIE-GENERATION MODEL
ANALYST'S MANUAL

September 1981

John B. Abell
Michael J. Konvalinka

Prepared pursuant to Department of Defense Contract MDA903-81-C-0166 (ML102). Views or conclusions contained in this document should not be interpreted as representing the official opinion or policy of the Department of Defense. Except for use for Government purposes, permission to quote from or reproduce portions of this document must be obtained from the Logistics Management Institute.

DTIC
ELECTE
FEB 23 1982
S
B

LOGISTICS MANAGEMENT INSTITUTE
4701 SANGAMORE ROAD
WASHINGTON, D.C. 20016

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited

PREFACE

This volume is the last of six volumes that describe the LMI Sortie-Generation Model System. Volume I, Executive Summary, discusses the problem the system is designed to address and provides an overview of the principal parts of the system. Volume II, Sortie-Generation Model User's Guide, provides sufficient information to allow a user to run the Sortie-Generation Model (SGM). Volume III, Sortie-Generation Model Analyst's Manual, describes the mathematical structures, derivations, assumptions, limitations, and data sources of the system at a very detailed level. Volume IV, Sortie-Generation Model Programmer's Manual, specifies the details of the computer programs, file structures, job control language, and operating environment of the system. Volume V describes the maintenance subsystem and explains the construction of the maintenance input file to the SGM. Volume VI describes the spares subsystem and shows a user how to build the spares file that is used by the SGM.

Potential users are cautioned that no volume is intended to provide, by itself, all of the information needed for a comprehensive understanding of the operation of the SGM.

RE: Classified References, Distribution Unlimited.
No change in distribution limitation per Lt. Col. Nimitz, OASD(MR&L)

Accession For
NTIS GRA&I ✓
DTIC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or Special
A

DTIC COPY INSPECTED

ACKNOWLEDGMENTS

We are indebted to Professor Edward J. Ignall of Columbia University for his many imaginative contributions to the formulation of the Sortie-Generation Model; to Professor Louis W. Miller of the Wharton School, University of Pennsylvania, who first suggested the basic structure of the multiple-server queueing system incorporated in the model; to Professor John A. Muckstadt of Cornell University for his helpful suggestions early in the model's development; and to our colleague at LMI, Mr. David L. Goodwin, who contributed greatly to the software development.

TABLE OF CONTENTS

LIST OF FIGURES

VOLUME III

SORTIE-GENERATION MODEL ANALYST'S MANUAL

1. MODEL DESCRIPTION

INTRODUCTION

The Sortie-Generation Model (SGM) is a hybrid analytic/simulation model that estimates the expected maximal number of sorties that can be flown by a specified type of tactical fighter aircraft in a wartime scenario. This estimate is based on aircraft characteristics, maintenance manpower and recoverable spares levels, and user inputs that describe the scenario of interest.

The purpose of this volume is to describe the mathematical structures, algorithms, and assumptions that underlie the SGM's estimates of sortie-generation capability. We also discuss, in Chapter 5, our efforts to validate the SGM.

In this chapter, we simply repeat, for the reader's convenience, our general description of the SGM that is also included in Volume II, Sortie-Generation Model User's Guide.

THE STRUCTURE OF THE MODEL: STATES AND PROCESSES

The SGM consists of a collection of aircraft states, processes that cause transitions between states, and logic that governs those processes. The SGM simulates the transition of aircraft between these states throughout a daily flying schedule that is specified by the user. The definitions of the states, the logic of the state transitions, and the interaction of these transitions with the flying schedule determine the basic structure of the SGM.

Aircraft States

There are five aircraft states in the SGM:

1) Mission-capable

2) Maintenance

3) Not mission-capable, supply (NMCS)

4) Combat loss

5) Reserve

These states are mutually exclusive and collectively exhaustive; i.e., every aircraft is in one and only one state. The states are described below.

Mission Capable. An aircraft is considered mission-capable if it is capable of flying a combat mission. It is not mission-capable if it is undergoing essential corrective maintenance, or is missing a mission-essential part. There is no explicit representation in the SGM of aircraft that are partially mission-capable.

Maintenance. An aircraft is in maintenance status if it requires unscheduled, on-aircraft repair that is essential to the performance of its mission. This repair may or may not be due to failure of a part; however, in this model, an aircraft is not allowed to enter maintenance until all needed parts have been obtained from supply or repair.

NMCS. An aircraft is not mission-capable, supply if the aircraft is missing an essential part. In the SGM, only mission-essential Line Replaceable Units (LRUs) can cause an aircraft to become NMCS.

Combat Loss. An aircraft is counted as a combat loss if it does not return from a sortie. Once an aircraft has been lost it can never be recovered. Battle-damaged aircraft that return from a sortie are not considered in this model.

Reserve. Reserve aircraft consist of mission-capable aircraft that are used to replace combat losses. The user specifies an initial number of aircraft that are held in reserve at the beginning of the scenario; these reserve aircraft replace combat losses at the end of each day, until all

reserves have been exhausted. Aircraft are allowed to leave this reserve state, but no aircraft can enter it.

Processes - Transitions Between States

There are eight processes in the SGM which cause transitions between aircraft states:

1) Ground aborts
2) Breaks
3) Aircraft repairs
4) Parts demands
5) Parts repair
6) Cannibalization
7) Attrition
8) Commitment of reserves

Figure 1-1 depicts the relationships among the various states and processes.

THE LOGIC OF THE MODEL: EVENTS AND THEIR SEQUENCING

The events that occur in the SGM are related to a flying schedule with user-specified characteristics. The flying schedule consists of a number of periods or cycles each of which is divided into three segments of lengths T_L, T_F, and T_W, respectively. During the last period, the T_W segment is replaced by an overnight recovery period. The user specifies the first and last take-off times of the day; the time, T_L, which is the average minimal length of time required to launch a sortie given a mission-capable aircraft; the sortie length, T_F; and the number of periods per day. The time, T_W, is then computed by the SGM program. The flying schedule is the same each day except for the number of aircraft to be flown each period, which the user can vary. A typical flying schedule is portrayed in Figure 1-2.

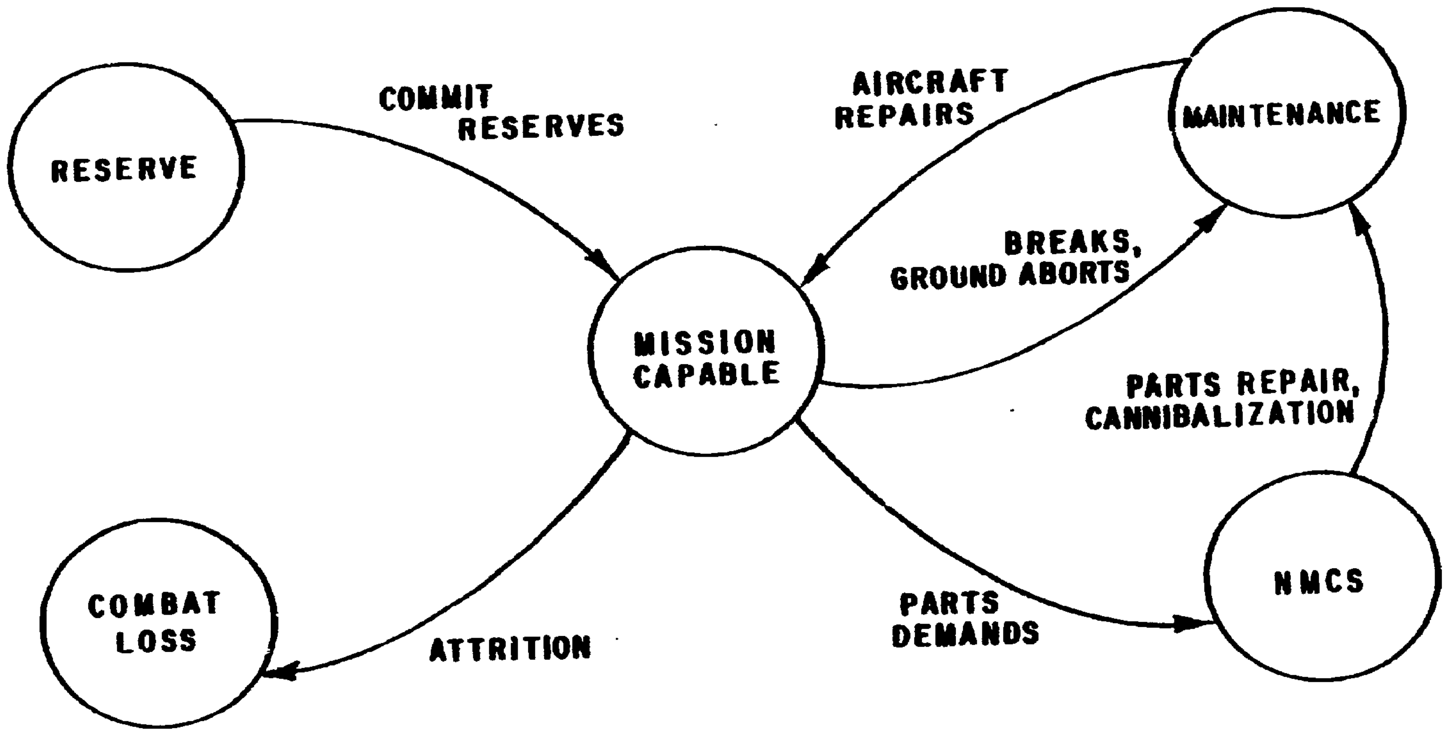
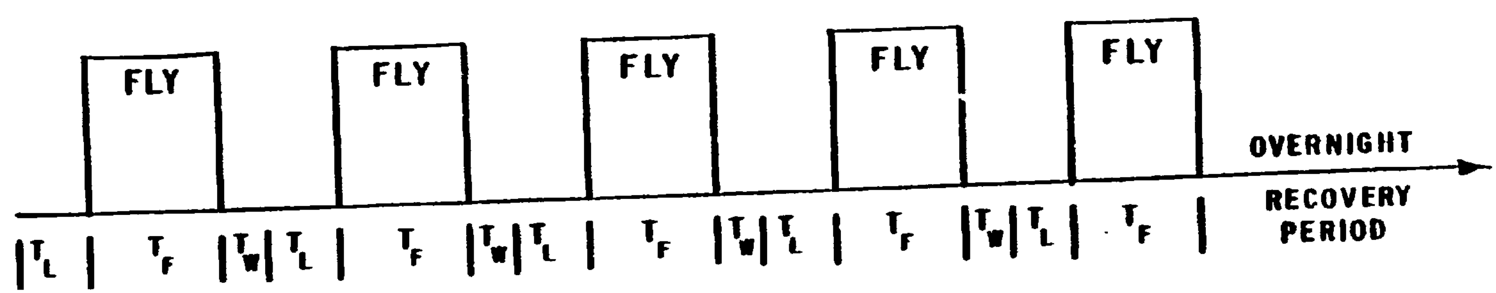


FIGURE 1-1
STATES AND PROCESSES



T_L = MINIMAL TIME TO LAUNCH GIVEN A MISSION-CAPABLE AIRCRAFT

T_F = SORTIE LENGTH

T_W = WAITING TIME (FIXED BY NUMBER OF PERIODS. FIRST AND LAST TAKEOFF TIMES, T_L AND T_F)

FIGURE 1-2

A TYPICAL FIVE-PERIOD FLYING DAY

Figure 1-3 portrays two segments of a flying day; the flying period on the left is intended to be typical and the one on the right to be the last period of the day. The events that occur in the SGM are denoted by circled numbers placed under the figures at the appropriate positions on the time line. Each of those events is described here.

Event 1

All mission-capable aircraft are prepared for launch. Any aircraft that is not mission-capable at this time (i.e., T_L before takeoff) cannot be flown during this cycle because, by definition, T_L is the minimal time required to launch an aircraft that is mission-capable.

Event 2

Aircraft that are repaired during the period of length T_L leave maintenance and become mission-capable but are not available for flight during this cycle.

Event 3

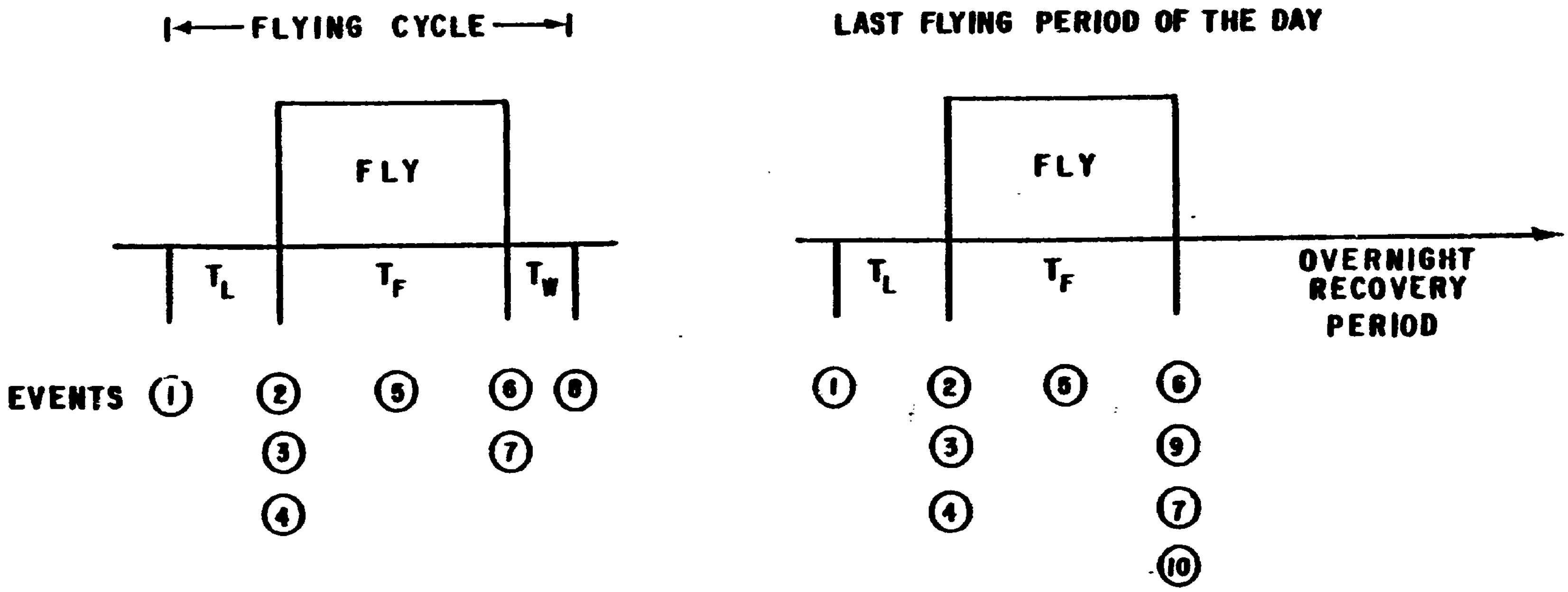
All aircraft that were prepared for takeoff are subjected to the probability of ground abort. A ground abort is defined as an unsuccessful attempt by an aircrew to fly an aircraft. The aborted aircraft enter maintenance. No parts demands are generated by ground aborts.

Event 4

The remaining aircraft that were prepared for takeoff fly sorties.

Event 5

Each aircraft that flies is subjected to the probability of attrition and, for each combat loss, an aircraft is deducted from the current strength of the organization.



T_L = MINIMAL TIME TO LAUNCH GIVEN A MISSION - CAPABLE AIRCRAFT

T_F = SORTIE LENGTH

T_W = WAITING TIME (FIXED BY NUMBER OF PERIODS, FIRST AND LAST TAKEOFF TIMES, T_L AND T_F)

FIGURE 1-3

SGM FLYING CYCLE

Event 6

Aircraft that are repaired during the period of length T_F leave maintenance and become mission-capable.

Event 7

Each aircraft returning from flight is subjected to the probability of break, i.e., the probability of requiring essential corrective maintenance prior to flying another combat mission. At the same time, parts demands are generated. Demands that can be filled from stock on-hand result in issues of that stock. Demands that cannot be filled from stock and cannot be satisfied by cannibalization from aircraft that are NMCS result in additional aircraft becoming NMCS.

Event 8

Aircraft that are repaired during the period of length T_W leave maintenance and become mission-capable.

Event 9

This event occurs only after the last flight of the day. It accounts for the parts repair process by subjecting each part in repair to the probability that the repair was completed during the preceding 24 hours. Remaining parts shortages are consolidated on as few aircraft as possible. If the consolidation results in fewer NMCS aircraft than before, the aircraft leaving NMCS status enter maintenance at this time.

Event 10

This event also occurs only after the last flight of the day. Combat losses may be replaced by available reserve aircraft, if the user so specifies. Any remaining reserve aircraft after losses have been replaced are committed according to user specification in the scenario input parameter. If reserves are to be used only as attrition fillers, then any remaining reserve

aircraft are left in the reserve pool; thus, the UE for the scenario will never increase. If the user has selected the reserve augmentation mode, then all reserve aircraft will be committed when they become available; hence, the UE for the scenario may actually increase.

THE REPAIR PROCESS

The entry of an aircraft into maintenance results from a ground abort or a "break" during a sortie. In either case, following the ground abort or sortie, the aircraft is subjected to a sequence of random draws that determines the subset of work centers that will be involved in the maintenance on that aircraft. A work center is a set of maintenance personnel with a particular skill. Examples of work centers are the structural repair shop, the hydraulic shop, and the automatic flight control system shop.

In the construction of the maintenance data base that supports the SGM, the aircraft repair times for all work centers involved in the repair of the aircraft are measured from the time of the ground abort or landing of the aircraft. For each work center involved in the repair, a random draw is made from an exponential distribution of repair time for that work center. The mean of that distribution is the reciprocal of the service rate contained in the maintenance data base for the work center in question. All work centers involved in the repair are assumed to work on the aircraft simultaneously; thus, the recovery time of the aircraft is simply the longest of the repair times for all the work centers involved in the recovery of that particular aircraft.

In the SGM, once the aircraft leaves maintenance and becomes mission-capable again, it loses its identity and is counted simply as another aircraft in the mission-capable pool.

CHAPTER 2. AIRCRAFT MAINTENANCE: GROUND ABORTS, AIRCRAFT BREAKS, AND AIRCRAFT REPAIRS

INTRODUCTION

In the SGM, aircraft enter maintenance in one of two ways, as a result of a ground abort or as a result of flying a sortie. A ground abort is an attempt by an aircrew to fly an aircraft, unsuccessful due to a malfunction or condition requiring corrective maintenance prior to flight. An aircraft break is the result of a malfunction during flight or discovery of a condition after flight that requires corrective maintenance prior to further flight. Thus, each aircraft scheduled for flight or returning from flight is subject to entering maintenance.

In this chapter, we discuss the algorithms used by the SGM to model aircraft ground aborts, malfunctions or "breaks" during flight, and the repairs of aircraft by maintenance work centers.

GROUND ABORTS

The ground abort algorithm uses the number of mission-capable aircraft scheduled for flight as the first parameter of a binomial distribution, the second parameter being the probability of ground abort. With these two parameters given, a random draw is made to determine the value of a binomial random variable designating the number of ground aborts. This number is passed to an algorithm that determines the work centers in the maintenance organization that must perform maintenance on each of the aborted aircraft. The ground abort algorithm then reduces the number of mission-capable aircraft that will be flown in the current flying period by the number of ground aborts.

The next step in the process is to determine the work centers involved in the repair of the aborted aircraft. This is accomplished by the sequential

sampling algorithm discussed in the section entitled <u>Distributing Maintenance Workload</u>.

No part demands are generated by ground aborts. All part demands are generated by aircraft breaks that occur after flight. The user is able to specify a ground abort rate that remains constant throughout the scenario or one that varies from day to day.

AIRCRAFT BREAKS

The user specifies the aircraft break rate, i.e., the probability that an aircraft returning from flight needs to undergo unscheduled, corrective maintenance prior to further flight. The aircraft break algorithm works essentially the same as the ground abort algorithm. A binomial random variable is generated from a distribution of which the first parameter is the number of aircraft returning from flight at the end of a flying period and the second parameter is the aircraft break rate. This model assumes that breaks among aircraft are stochastically independent. After determining the number of broken aircraft, the SGM determines the numbers and types of component demands that resulted from these broken aircraft. The number of NMCS aircraft is then computed, and any remaining broken aircraft which are not NMCS are passed to the sequential sampling algorithm discussed in the next section. The parts demands and NMCS computations are described in Chapter 3.

Use of maintenance data from the Air Force's maintenance data collection system results in estimates of break rates by work center that are higher than reality. Therefore, work center break rates (i.e., the proportion of broken aircraft on which work is done by individual work centers) in the SGM are treated as conditional probabilities given an aircraft break, rather than the conditioning event being an aircraft sortie. Thus, the aircraft break rate is

explicitly specified by the user as an input variable rather than being inferred from the work center break rates.

DISTRIBUTING MAINTENANCE WORKLOAD

Once it is determined that an aircraft requires maintenance, because of either a ground abort or an aircraft break, this algorithm determines the particular work centers where the aircraft requires repair. It does this by generating a random sample from among all the possible ways an aircraft can break into maintenance. The sampling is consistent with the work center break rates.

This algorithm is a sequential sampling technique that deals iteratively with each work center. Given that an aircraft has broken into at least one work center, a random number is drawn for each work center, beginning with the first, to answer the questions:

1. Did the aircraft break into this work center?
2. Did the aircraft break into any other work centers which have not been considered yet?

If the answer to the first question is yes, the aircraft is "marked" to indicate that it has broken into this work center. The answer to the second question determines whether the algorithm is terminated for this aircraft, or continues with the next work center.

The key assumption in the formulation of the algorithm is that breaks into different work centers are independent, i.e., a break into one work center does not affect the probability of breaking into another work center. This independence assumption simplifies modeling the probabilities associated with the above questions and allows the algorithm to work in an iterative fashion without "remembering" the outcomes for previous work centers.

The remainder of this section provides a detailed description of the algorithm and a proof that the random samples generated are consistent with

the probability distribution of the possible ways an aircraft can break into maintenance.

Given:

- A collection of N work centers, labeled 1, 2,..., N.
- An aircraft breaks into a least one of these work centers.
- B(j) for j = 1, 2,...,N. B(j) is the probability that an aircraft breaks into work center j. These probabilities are assumed to be independent.

Determine:

- A random sample of the work centers where the aircraft requires repair.

Approach

This algorithm is an iterative sampling technique that deals sequentially with each work center j: j = 1, 2,..., N, until termination is indicated. The jth iteration proceeds as follows: Assume that the aircraft has broken into at least one of the work centers in the set {j, j + 1, j + 2,..., N}. (Clearly this assumption holds for the first iteration, j = 1.) Then one of the following situations must exist:

Event A_j : The aircraft has broken into work center j, but has not broken into any of the work centers in the set {j + 1, j + 2,..., N}.

Event B_j : The aircraft has broken into work center j and also broken into at least one of the work centers in the set {j + 1, j + 2,..., N}.

Event C_j : The aircraft has not broken into work center j, but it has broken into at least one of the work centers in the set {j + 1, j + 2,..., N}.

Given the probabilities of these events, a random sample can be generated to determine which situation applies. The results of this sample determine the answers to the following questions:

1. Did the aircraft break into work center j?

2. Did the aircraft break into any of the remaining work centers, $\{j + 1, j + 2, \ldots, N\}$?

The first question determines whether the aircraft is marked as having broken into work center j; the second question determines whether the algorithm terminates for this particular aircraft, or continues with the next work center. The algorithm proceeds as follows:

Step 0: Initialize probabilities for each work center.

Calculate the following probabilities for each work center, $j=1,2,\ldots,N$;

PMARK(j) = Probability that an aircraft breaks into work center j, given it has broken into at least one of the work centers in $\{j, j + 1, \ldots, N\}$,

$$= B(j) / \{1 - \prod_{k=j}^{N} [1-B(k)]\};$$

PSTOP(j) = Probability that an aircraft has broken into work center j but has <u>not</u> broken into any of the work centers in the set $\{j + 1, j + 2, \ldots, N\}$, given that the aircraft has broken into at least one of the work centers in $\{j, j + 1, \ldots, N\}$,

$$= \text{PMARK}(j)\{\prod_{k=j+1}^{N} [1-B(k)]\}.$$

(Note: PSTOP(N) = PMARK(N) = 1)

PMARK(j) and PSTOP(j) represent points in the cumulative distribution function of the events: A_j, B_j, and C_j. Specifically, PSTOP(j) equals the probability of event A_j for an aircraft which is given to have broken into at least one of the work centers $\{j, j + 1, \ldots, N\}$ and PMARK(j) equals the sum of the probabilities of A_j and B_j. In this algorithm, a uniform (0, 1) random draw is compared with these probabilities to determine a random sample from among the three possible events for work center j.

Note that the above formulae are based on the assumption that the break probabilities, $\{B(j)\}$, are independent. Also, PSTOP(N) equals 1, since

A_N is the only possible event for work center N; i.e., there are no more work centers in which the aircraft may require repair.

The break probabilities, $\{B(j)\}$, for the various work centers are assumed to remain constant throughout the scenario; consequently, Step 0 needs to be executed only once at the beginning of the simulation.

The main loop of this algorithm consists of three steps which are executed for each work center, $j = 1, 2, \ldots, N$, until termination is indicated. The steps are:

Step 1: Generate a uniform (0, 1) random number, U.

This random draw determines which of the three possible events has occurred for work center j. The determination is made in Steps 2 and 3 and is based on where the random draw falls in the cumulative distribution of these events:

$$0 < U \leq \mathrm{PSTOP}(j) \text{ implies Event } A_j$$

$$\mathrm{PSTOP}(j) < U \leq \mathrm{PMARK}(j) \text{ implies Event } B_j$$

$$\mathrm{PMARK}(j) < U \leq 1 \text{ implies Event } C_j$$

Step 2: Determine if the aircraft has broken into work center j.

If $U > \mathrm{PMARK}(j)$, then increment the work center index j, and return to Step 1.

If $U \leq \mathrm{PMARK}(j)$, then "mark" the aircraft as having broken into work center j.

If the random draw indicates Event C_j, i.e., $U > \mathrm{PMARK}(j)$, then the aircraft did not break into work center j, but it does require repair in at least one of the remaining work centers, $\{j + 1, j + 2, \ldots, N\}$. Therefore, the algorithm continues with the next work center.

If Event C_j is not indicated, then either Event A_j or Event B_j occurs. Both of these situations imply that the aircraft has broken into work

center j. A further test will be made in Step 3 to distinguish between these two remaining possibilities.

Step 3: Determine if the aircraft has broken into any of the remaining work centers.

If U > PSTOP(j), increment the work center index j, and return to Step 1.

If U $\leq$ PSTOP(j), stop.

If Event A_j or Event B_j was indicated in Step 2, another test is needed to distinguish between these two events. If the random draw indicates Event B_j, i.e., U > PSTOP(j), the aircraft has broken into one of the remaining work centers, $\{j + 1, j + 2, \ldots, N\}$. Hence, the algorithm continues with the next work center.

If Event A_j is indicated, then work center j is the last work center in which this aircraft requires repair; and the algorithm is terminated for this aircraft. Note that the algorithm can only terminate at a work center in which the aircraft requires repair, thus ensuring that it always breaks into at least one work center. The algorithm is also guaranteed to stop eventually since PSTOP(N) = 1. A flow chart of the algorithm is shown in Figure 2-1.

SIMULATING AIRCRAFT REPAIRS

This section describes an algorithm for simulating the number of aircraft repaired in a work center during a time period of specified length. In the Sortie-Generation Model, aircraft repair is modeled as a multiple-server queuing process. A work center consists of a specified number of servers or crews that perform unscheduled maintenance on aircraft that break into the work center. Each crew within a work center has an independent and identical exponential service-time distribution. No crew is ever idle while an aircraft is waiting for service, i.e., if there is a queue, then all crews are busy.

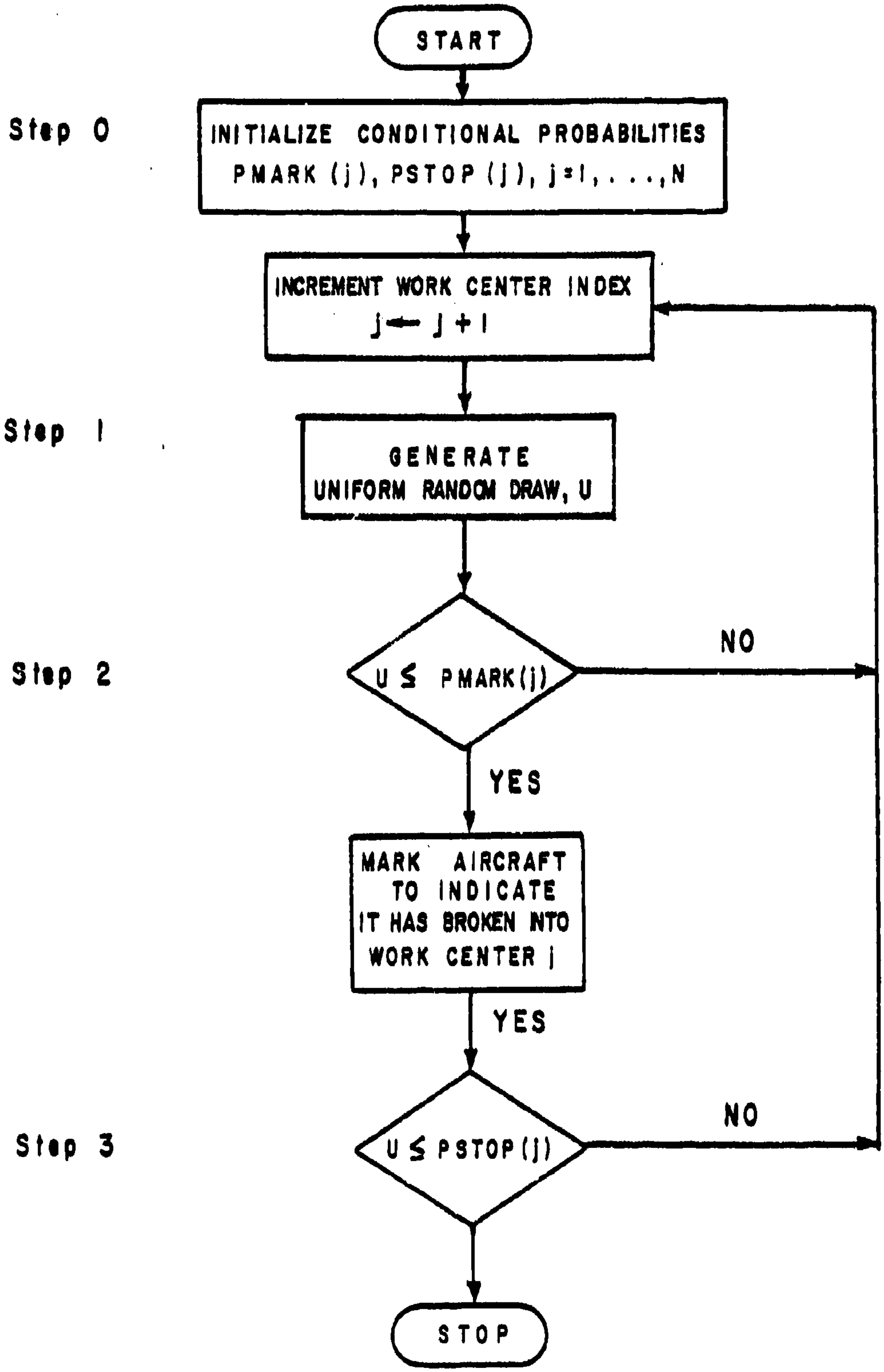


FIGURE 2-1 FLOW CHART OF MAINTENANCE WORKLOADING ALGORITHM

In addition, aircraft are assumed to enter the work center only at the beginning or end of a period. Under these assumptions, the number of aircraft repaired in the work center is a random variable whose probability distribution is dependent only on:

1. the number of aircraft in the work center at the start of the period,
2. the number of crews,
3. the service rate for these crews, and
4. the length of the period.

Note that the number of aircraft repaired does not depend on how long any aircraft has been in the work center prior to the start of the repair period, nor on the queuing discipline (order of service) employed in the work center. The purpose of this algorithm is to generate a random sample of the number of aircraft repairs that is consistent with the probability distribution of the corresponding random variable. For a detailed discussion of the output from this type of queuing process, see [14].

The sampling technique employed by this algorithm is determined by the state of the system at the start of the period. If the number of aircraft in a work center does not exceed the number of servers in the work center, then all aircraft are being serviced at the start of the period, and the number of aircraft repaired is a binomially distributed random variable. A random sample is generated from a binomial distribution for which the first parameter is the number of aircraft in the work center at the start of the period and the second parameter is the probability that any single aircraft will be completed during the period.

If one or more aircraft are waiting for service at the start of the period, the number of aircraft repaired during the period is not binomially distributed. In this case, all crews will be busy for at least some portion

of the repair period -- possibly the entire period. If all crews are busy throughout the period, the time between aircraft repairs in the work center is an exponentially distributed random variable. An approach to modeling this system is to generate samples from the exponential distribution of time between repairs. Each of these "inter-repair" times represents an aircraft repair and their sum indicates the elapsed time in the repair period. The sampling is continued until either the elapsed time exceeds the length of the repair period or enough aircraft are repaired so that a crew becomes idle. Once the first crew becomes idle, a binomial random sample can be generated to determine how many of the remaining aircraft are repaired by the end of the period.

Given:

A = Number of aircraft in the work center at the start of the repair period. (A is a non-negative integer.)

r = Service rate (in aircraft per hour) of each server; it is the reciprocal of the mean of the exponential service-time distribution.

s = Number of servers or crews in the work center. (s is a positive integer.)

t = Length (in hours) of the repair period.

Determine:

R = Random sample of the number of aircraft repaired in the work center during the repair period. (R is an integer such that $0 \leq R \leq A$.)

Approach

The sampling technique used is determined by the state of the system at the start of the repair period:

Case 1. The number of aircraft in the work center at the start of the period does not exceed the number of servers, that is, $A \leq s$

In this case, each aircraft is being serviced by a repair crew at the start of the period. The service time for each crew is exponentially distributed; hence, the probability of an aircraft completing service during the period depends only on the length of the period and the service rate of the crew.

The probability that a particular aircraft is repaired during the period, given that it is in service at the start of the period is $1 - e^{-rt}$. Thus, each aircraft in the work center at the start of the period constitutes a Bernoulli trial, the probability of completing service being equal to $1 - e^{-rt}$. Since all crews are working independently, the random variable, total aircraft repairs, is equivalent to the sum of A independent, identical Bernoulli trials, i.e., a binomially distributed random variable. Hence, the sampling procedure for Case 1 consists of generating a random sample from a binomial distribution with parameters A and $1 - e^{-rt}$.

Case 2. One or more aircraft are waiting for service at the start of the period, i.e., $A > s$.

In this situation, all s crews will be busy for at least some portion of the repair period -- possibly the entire period. As long as a particular crew is busy, the time between aircraft repairs for that crew is distributed exponentially with parameter r during that busy period. Furthermore, if all s crews are busy for a given interval, the time between aircraft repairs for the entire work center is distributed exponentially with parameter rs during that interval [6:284]. This exponential distribution will apply until one of the crews completes a repair and finds no more aircraft to fix, i.e., when exactly s-1 aircraft remain, or equivalently, when A-(s-1) aircraft have been repaired.

Thus, the approach for Case 2 is to generate a sequence of random draws, T_1, T_2,..., from an exponential distribution with parameter rs to

simulate the inter-repair time between successive aircraft repairs in the work center. Each of these draws represents the completion of an aircraft repair and the sum of, say, k of them indicates the elapsed time required for k aircraft repairs. This sequence is continued until one of the following situations occurs:

1. The total elapsed repair time exceeds the length of the period, i.e., $T_1 + T_2 + \ldots + T_k > t$, or
2. A server becomes idle, i.e., exactly A-(s-1) aircraft have been repaired.

If situation 1 occurs first, then the simulated number of aircraft repairs is the number of repairs completed before the last exponential draw caused the sum of inter-repair times to exceed the length of the repaired period, i.e., $R = k-1$.

If situation 2 occurs first, then A-(s-1) aircraft have been repaired in time $T_I = T_1 + T_2 + \ldots + T_{A-(s-1)}$, and s-1 aircraft are still being serviced. Since $T_I < t$, there is still a time interval $(t-T_I)$ left in which to repair these remaining aircraft. Due to the memoryless property of the exponential distribution of service times, we now have the same situation as when all remaining aircraft are in service at the beginning of a time interval. Hence, we can generate a random sample, R_I, from a binomial distribution with parameters s-1 and $1-e^{-r(t-T_I)}$ to determine how many of the remaining s-1 aircraft are repaired, and set R equal to $R_I + A-(s-1)$. This sequential sampling process is a combination of several well known techniques for generating random variates [2:203-204, 224].

Using the inverse transformation method, an inter-repair time sample from an exponential distribution with parameter rs is given by:

$$T_i = -(1/rs)\log_e(U_i),$$

where U_i is a random draw from a uniform (0, 1) distribution. Thus, total elapsed repair time can be written in the form:

$$
\begin{aligned}
&T_1 + T_2 + \ldots + T_k \\
&\quad = [-(1/rs)\log_e U_1] + [-(1/rs)\log_e U_2] + \ldots + [-(1/rs)\log_e U_k] \\
&\quad = -(1/rs)\log_e (U_1 U_2 \ldots U_k),
\end{aligned}
$$

where U_1, $U_2, \ldots,$ U_k are independent samples from a uniform (0, 1) distribution. The algorithm termination test can be restated as follows:

$$
\begin{aligned}
& & T_1 + T_2 \ldots + T_k &> t, \\
&\text{which implies} & -(1/rs)\log_e (U_1 U_2 \ldots U_k) &> t, \\
&\text{or} & U_1 U_2 \ldots U_k &< e^{-rst}.
\end{aligned}
$$

Thus, the algorithm generates a sequence of uniform (0, 1) samples until their cumulative product no longer exceeds e^{-rst}.

If A-(s-1) aircraft are repaired before the end of the period, then T_I, the time at which the first server became idle, is easily calculated from this cumulative product:

$$
\begin{aligned}
T_I &= T_1 + T_2 + \ldots + T_{A-(s-1)} \\
&= -(1/rs)\log_e \{U_1 U_2 \ldots U_{A-(s-1)}\}
\end{aligned}
$$

Algorithm 1: Work Center Queue is Empty, i.e., $A \leq s$.

Step 1: Generate the random sample r from the appropriate binomial distribution.

(a) Generate a random sample from a binomial distribution with parameters A and $1-e^{-rt}$.

Algorithm 2: Work Center Queue is Not Empty, i.e., $A > s$.

Step 0: Initialize aircraft repaired and total elapsed repair time to zero.

(a) Set R = 0 and P = 1.

Step 1: Generate next aircraft inter-repair time and update elapsed repair time.

(a) Generate random draw U from uniform (0, 1) distribution.

(b) Set $P = P \cdot U$.

Step 2: Determine if this next aircraft was repaired before the end of the repair period.

(a) If $P < e^{-rst}$, the algorithm is terminated; R contains number of aircraft repaired during this repair period.

(b) If $P \geq e^{-rst}$, set $R = R + 1$; continue with Step 3.

Step 3: Determine if enough aircraft have been repaired so that a server has just become idle.

(a) If $R < A - (s-1)$, return to Step 1.

(b) If $R = A - (s-1)$, continue with Step 4.

Step 4: Compute the time at which the first server became idle.

(a) Set $T_I = -(1/rs) \cdot \log_e(P)$.

Step 5: Determine how many of the remaining aircraft are repaired before the end of the period.

(a) Generate a random sample R_I from a binomial distribution with parameters $(s-1)$ and $1-e^{-r(t-T_I)}$.

(b) Return $R = R_I + A-(s-1)$ as the number of aircraft repaired.

3. AIRCRAFT SPARES: PARTS DEMANDS, PARTS REPAIRS AND CANNIBALIZATION

INTRODUCTION

In this chapter, we discuss the algorithms used by the SGM to model demands, repairs, and cannibalization of aircraft parts. The SGM abstracts these processes to the extent that it generates demands for parts without regard to the work centers involved in aircraft repair. Nevertheless, the parts demand rates and maintenance workload distribution in the model are consistent with the input data derived from the various Air Force data sources used and are consistent with each other except within an aircraft tail number. These relationships are explained in detail in the discussion that follows.

The principal purpose of the algorithms discussed here is to constrain sortie production by estimating accurately the number of aircraft that are in "not mission capable, supply" (NMCS) status at any point in time. The problem is to make that constraining effect consistent with the original budget level established by the user in the Interactive Budget Allocation Program so that the sortie-generation profile produced by the SGM reflects the effect of that budget level realistically.

PARTS DEMANDS

Initialization

At the start of the simulation, the SGM initializes the supply posture of each part, i.e., it determines the number of parts of each type in resupply. By resupply we mean base repair, depot repair (including retrograde shipment), and in transit from depot to base (order and ship pipeline). In

the SGM, parts in transit are modeled as being in depot repair. The initialization is done by generating a Poisson random variate for each component from a distribution for which the mean is the expected number in resupply. If this number is not zero, it becomes the first parameter of a binomial random variate that is then generated for each part type to determine the number of parts in depot repair and in transit from depot to base. The rest of the parts are then assigned to base repair. The second parameter of the binomial random variate denoting the number to be assigned to depot repair and in-transit is given by

$$p = \frac{(1 - f)(t_w + t_o)}{ft_b + (1 - f)(t_w + t_o)} ,$$

where f = the base repair fraction (1 - NRTS rate),

t_w = depot delay time,

t_o = order-and-ship time, and

t_b = base repair time.

The NMCS Computation

The SGM assumes perfect cannibalization when translating the number of parts of each type in resupply to the number of NMCS aircraft. The number of NMCS aircraft is given by

$$N = \max_i \left\{ \left\lceil \frac{\max(n_{r,i} - n_{s,i}, 0)}{q_i} \right\rceil \right\} ,$$

where i denotes the ith part type,

$n_{s,i}$ = the stock level of part type i,

$n_{r,i}$ = the number of part type i in resupply,

q_i = the quantity per aircraft of part type i, and

$\lceil x \rceil$ = the smallest integer greater than or equal to x.

Thus, the number of NMCS aircraft is simply the greatest number of shortages of any one part type.

The assumption of perfect cannibalization is not realistic; however, it is felt to be a much better approximation to reality in a wartime surge environment than an assumption of a random distribution of parts shortages among aircraft.

Postflight Parts Demands

All parts demands are assumed to occur after flight. Clearly, parts demands also occur between flights, after ground aborts, during preflight, and at other times, but this simplifying assumption has a negligible effect on the SGM's estimate of sortie-generation capability and dramatically reduces computational time by minimizing the number of times during each simulated 24-hour period that parts demands have to be computed. At the end of each flying period, the computational algorithm translates the number of "broken" aircraft into the number of demands for each part type. In turn, the number of demands is translated into NMCS aircraft, in the manner described above.

The number of parts demands, given a specified number of broken aircraft, is computed as follows.

Let N = a random variable denoting the total number of parts demands generated during a flying period,

$P(B)$ = the probability that an aircraft breaks, given that it flies a sortie and returns,

n = the number of different part types,

q_i = the quantity per aircraft of part type i,

$P(D_i)$ = the probability of a demand for a component of type i, given that it is installed on an aircraft that flies a sortie and returns,

X_i = a random variable denoting the total number of demands in a flying period for part type i, and

k = the number of broken aircraft.

We assign to X_i a binomial distribution with parameters kq_i and $P(D_i)/P(B)$. Then,

$$N = \sum_{i=1}^{n} X_i .$$

The demands among parts of different types are assumed to be independent. Therefore, the expected value of N is given by

$$E(N) = E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n}\left[E(X_i)\right]$$

$$= \sum_{i=1}^{n}\left[kq_i P(D_i)/P(B)\right] = \frac{k}{P(B)} \sum_{i-1}^{n} q_i P(D_i);$$

similarly, the variance of N is equal to

$$V(N) = V\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n}\left[V(X_i)\right]$$

$$= \sum_{i=1}^{n}\left\{kq_i\left[P(D_i)/P(B)\right]\left[1 - P(D_i)/P(B)\right]\right\}$$

$$= \frac{k}{P^2(B)} \sum_{i=1}^{n} q_i P(D_i)\left[P(B) - P(D_i)\right].$$

If $E(N) \geq 20$, we invoke the Central Limit Theorem and approximate the distribution of N with a normal distribution (with continuity correction) whose mean and variance are as shown above. We also ensure that

$$0 \leq N \leq k \sum_{i=1}^{n} q_i .$$

If $E(N) < 20$, we approximate its distribution with a Poisson random variate with parameter E(N).

Given a value of N, it remains to distribute the N demands over the n part types. This is done as follows:

1. Partition the unit interval into n subsets, with subset i of size

$$p_i = q_i r_i / \sum_{j=1}^{n} q_j r_j, \quad i = 1, 2, \ldots, n,$$

where r_i = hourly demand rate of part i times sortie length

2. Generate N random variates that are uniformly distributed on the unit interval; use these values to select the N parts that experience demands.

Since n is large and p_i is small for any i, the approximation is excellent.

The only time at which repairs of parts can be completed is after the last flight of the day. After determining parts demands that occur as a result of the last flying period, the SGM determines the numbers and types of parts that are received from base and depot repair. This process is described in the section that follows.

Also, after the last flight of the day, another set of random variates is generated to decide how many parts of each type that failed that day are sent to the depot for repair and how many are repaired at the base. For each part, a random variate is generated from a uniform distribution on the unit interval and compared to the not-repairable-this-station (NRTS) rate for that part type. Based on these outcomes, two arrays are updated, one reflecting the number of each part type in base resupply and the other the number in depot resupply.

Finally, the number of current backorders for each part type is used to update the number of NMCS aircraft.

PARTS REPAIRS

Once each day, after the last flying period, parts are allowed to complete repair and be installed in NMCS aircraft or returned to the shelf, ready for issue. The distributions assigned to both depot and base repair times are exponential with means derived from the D041 system. The exponential assumption is equivalent to the assumption of an arrival rate of parts from repair directly proportional to the number of parts of each type in repair. The exponential assumption was made because of the distribution's memoryless property, thus allowing the SGM to account only for the numbers of parts of each type in resupply at the depot and at the base, and because no need then exists to keep track of the time at which a part entered resupply. The assumption is unrealistic in scenarios in which transportation or repair capacity is seriously constrained but is reasonable in most other cases and provides dramatic reductions in running times over distributions requiring storage of repair starting times for individual parts.

In determining the number of repairs of each part type completed at the end of a simulated flying day, the SGM generates two random draws from binomial distributions for each part type in resupply, either at depot or base. The first draw determines the number returned from the depot; the second draw, the number from base repair. The first parameter of the binomial distribution involved is the number of parts of that given type in resupply, whether depot or base; the second parameter is the probability of return from repair (depot or base) during the 24-hour day just ending.

Subsequent to this sampling, the SGM updates the number of parts of each type in each echelon of repair, updates the backorders for all part types, and recomputes the number of NMCS aircraft.

4. ATTRITION AND AIRCRAFT RESERVES

INTRODUCTION

This chapter describes the SGM treatment of aircraft attrition due to combat losses and the replacement of these losses with available reserve aircraft.

COMBAT ATTRITION

For purposes of estimating sortie-generation capability, there are actually two dramatically different kinds of aircraft combat attrition. The first is simply the loss of an aircraft on a mission, i.e., its failure to return from a flight. The second kind involves the return of an aircraft to its home base coupled with the loss of the future use of the aircraft for a relatively long period of time, perhaps the entire length of the scenario of interest, owing to combat damage. The importance of the distinction lies in the fact that the combat-damaged aircraft typically yields a fairly complete range of spare parts for other aircraft (complete except for parts that were damaged and parts that are not interchangeable among aircraft). In addition, these aircraft require a different mix of maintenance skills and spares to repair than do aircraft that fly sorties and return without combat damage (see [5] and [9]). The SGM treats only the first type of combat attrition. We assume no combat-damaged aircraft ever return to the base.

The SGM implements combat attrition as a binomial sampling process. At the end of each sortie period, a random sample is generated from an appropriate binomial distribution to determine the number of combat losses among the aircraft which flew the mission. These combat losses are then subtracted from the current strength of the organization; however, each is counted as having flown a sortie for that particular period. The parameters of the binomial

distribution for determining the combat losses consist of a user-specified attrition probability and the number of aircraft that flew on a particular mission. Thus, the attrition of aircraft during a mission is implemented as a collection of independent Bernoulli trials with the attrition of one aircraft during a sortie having no effect on the possible attrition of other aircraft flying the same mission.

The attrition probability is specified by the user for each day of the flying scenario. Thus, the user can tailor the daily attrition rates to represent the effect of previously flown sorties. A typical wartime scenario would probably assume higher attrition rates during the initial stages of the conflict, declining to some lower rate as air superiority is achieved or tactics are adjusted. The attrition rates remain constant in the SGM during a flying day, i.e., each wave is subject to the same attrition probability; however, this limitation is for convenience of data input only. The SGM could be easily modified to use different attrition rates for each mission of the day.

It should be noted, of course, that the assumption of an attrition rate that is independent of the sorties generated thus far in the conflict is somewhat naive. Presumably, the more sorties flown to eliminate counter-air capability, the lower the attrition one might expect subsequently, i.e., an earlier sortie is somehow more important than a sortie flown later in the scenario. Thus, the attrition rate could reasonably be modeled as some function of total sorties generated thus far. This might also provide a better estimate of the sortie-generation capability for the specified logistic resources.

Including attrition in a flying scenario has a significant impact on the comparison of the sortie-generation capabilities inherent in different logistic resource mixes. It is characteristic of a constant attrition rate that the expected total number of sorties flown is equal to the total force size divided by the attrition rate. Thus, although increases in logistic resources can make significant differences in sortie-generation capability, the expected total number of sorties flown is a function of the attrition rate alone. Figure 4-1 illustrates this impact of attrition upon the sortie profile. Two profiles estimated from different resource mixes are shown. Obviously, the better mix yields a much higher sortie rate early in the conflict; however, as attrition takes a greater toll on the better mix, the sortie profiles cross and the second mix begins to produce more sorties. Thus, at the end of the 30-day period, both resource mixes have produced nearly the same total number of sorties, even though one resource mix is obviously better than the other.

RESERVE AIRCRAFT

Reserve aircraft consist of fully mission-capable aircraft that are used either to replace combat losses or to augment the organization's current aircraft strength during the flying scenario. These reserves represent replacement aircraft that are flown in each day. They are assumed to arrive in a fully mission-capable status, and once a reserve has been committed, it cannot return to reserve status, i.e., aircraft are allowed to leave the reserve state, but no aircraft can enter it.

At the start of each SGM simulation, the user specifies how reserves are to be committed throughout the scenario. Reserves can be used as "attrition fillers" or they can be used to augment the organization's current aircraft strength. If reserves are to be used as attrition fillers, they are committed on an as-needed basis to replace any combat losses that may occur during the

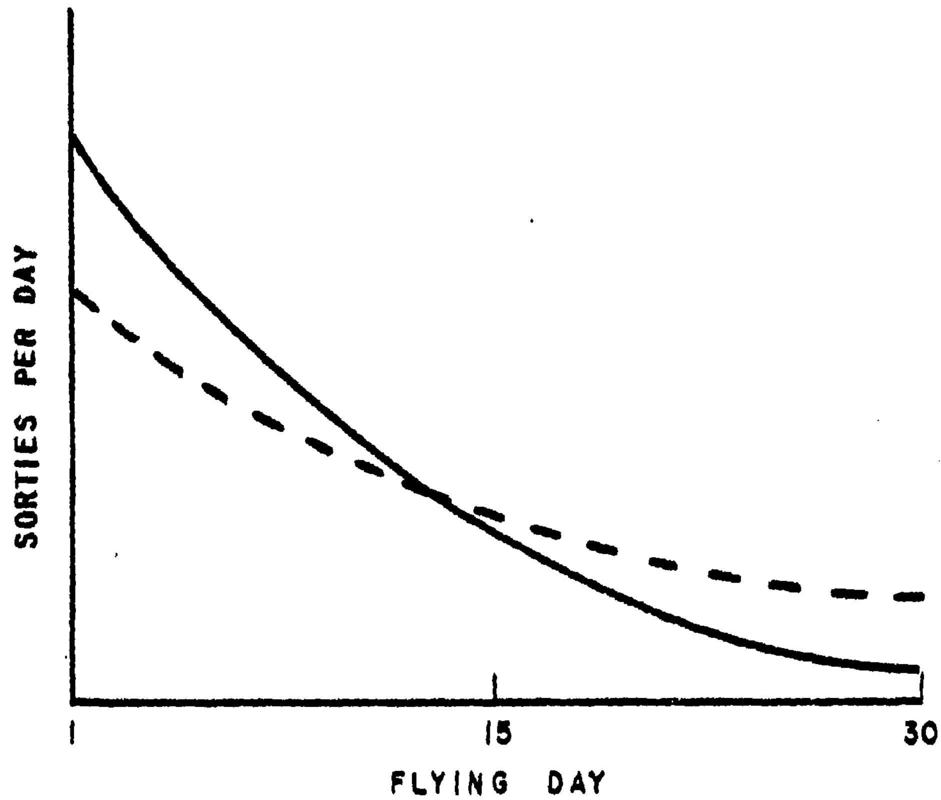


FIGURE 4-1 EFFECT OF ATTRITION ON SORTIE PROFILE

day. In this mode, not all reserves may be committed on the day they become available.

If the augmentation mode is selected, then all reserve aircraft are committed on the day they become available; hence, the size of the force may actually increase during the scenario. An example of this situation would be a scenario in which a squadron is deployed after the outbreak of a conflict to join a wing already fighting in-place.

After selecting the manner in which reserves are to be used, the user also specifies, for each flying day, the number of reserve aircraft that become available on that day. The reserves are committed in the following manner: if the user specifies that X reserve aircraft are available as of the nth day, then immediately after the last flight of the nth day, these X new reserves are added to the current reserve pool. Next, aircraft from this reserve pool are used to replace combat losses or augment the current aircraft strength. In either case, the earliest possible sortie that any of these X new reserves could fly, is the first sortie of the next (i.e., the (n + 1)st) day.

5. VALIDATION EFFORTS

INTRODUCTION

The purpose of this chapter is to describe the validation efforts which have been conducted in support of the Sortie-Generation Model (SGM). The Sortie-Generation Model is only one part of a system of models that includes the SGM Spares Subsystem and the SGM Maintenance Subsystem. These subsystems generate spares and maintenance data bases that are input to the SGM. In addition, the SGM has been designed to model a variety of flying scenarios for many different tactical aircraft weapon systems. For these reasons validation of the SGM has been a complex task. Extensive tests have been conducted; however, this validation effort is not complete. For the weapon systems and scenarios tested thus far, the SGM has provided consistent and reasonable estimates of sortie-generation capability.

The SGM validation effort has followed the general framework and employed many of the techniques described in references [12] and [15]. Our effort has been conducted in three phases. First, extensive testing was conducted to ensure the "correctness" of the software comprising the model. Next, analytic experiments and sensitivity analyses were performed to verify the validity of the model algorithms and approximations. The final phase was to examine the adequacy of the model in predicting "real system" behavior. The following sections of this chapter describe these phases in more detail.

SOFTWARE VERIFICATION

The first step in any validation effort for a large simulation model is to ensure that the model is performing as designed by verifying the correctness of the model's computer programs. Although the concept is quite

simple, the actual task of "debugging" a large software program is tedious. Any large, complex piece of software should always be used with caution. A large software program can perhaps never be considered completely bug-free; however, it should be designed and tested in a manner sufficient to instill confidence in the model user. For this reason, a great deal of effort was devoted to the design and testing of the SGM software to ensure program correctness.

Design goals in the development of the SGM included such software engineering techniques as modular design, structured programming, and extensive, built-in error checking. The model was written and debugged in small modules. Most SGM routines are one page in length and none are more than three pages. Although the SGM is written in FORTRAN, structured programming was enforced through the use of standard IF-THEN-ELSE constructs and looping operations. Finally, the SGM contains a great deal of what might be called "defensive programming". Extensive checks with error messages have been built into most of the SGM routines. A more detailed discussion of these SGM design goals is contained in Volume IV, SGM Programmer's Manual.

After the initial development of the SGM, an extensive testing phase was conducted. Each SGM module was read, reviewed, and tested by an "independent" programmer or analyst, i.e., someone other than the original designer and coder. An interesting and important by-product of this phase was the identification of computational bottlenecks in the model. As part of the testing procedure, we determined the amount of time spent in each of the model's various processes for a typical SGM run. Once the most time-consuming processes were identified, new algorithms were developed that resulted in significant improvements in the model's run time.

VERIFICATION OF MODEL ALGORITHMS AND APPROXIMATIONS

The second phase of the validation effort consisted of devising and conducting experiments to test the validity of the model's algorithms and approximations. The actual implementation of the model design involved the use of various approximations to achieve reasonable model running times and memory requirements.

The basic approach used in these tests was to design model scenarios with enough simplifying assumptions to allow us to calculate analytically what the expected answers should be. The model was run with these simplifying assumptions and the answers compared to the analytically predicted results. A simple example of this approach is the following experiment we ran to test the attrition portion of the model. It is characteristic of a constant attrition rate that the expected total number of sorties flown is equal to the total force size divided by the attrition rate. Thus, the model results can easily be tested by comparing the actual sorties flown with the predicted result.

Another test procedure was to compare the model results, using an algorithm or approximation, to another more detailed simulation of the same process. The alternative simulation used was either an entirely different model or the SGM itself with a more detailed implementation of the algorithm to be tested. These detailed implementations were adequate for testing purposes; however, they were too prohibitive in running time and memory requirements to provide any practical use. This phase was an attempt to answer the question, "Given the underlying assumptions of the model are true, does the SGM produce results that are consistent with those assumptions?"

During this phase, we also performed an extensive sensitivity analysis of the model's parameters to confirm its consistency over the range of interest.

This analysis involved a large number of SGM runs to identify those parameters to which the results were most sensitive.

PREDICTIVE VALIDATION

The final phase of our validation was to test the model's ability to provide a reasonable prediction of sortie-generation capability. Our first step was to compare the SGM with data from an actual Air Force sortie surge exercise, Salty Rooster. Next, we compared the SGM with other sortie models currently in use. We ran SGM comparison tests with the Air Force LCOM model. The remainder of this section describes the results of our Salty Rooster and LCOM tests.

Salty Rooster

Salty Rooster was a sortie surge exercise conducted at Hahn AB, Germany in April 1978. The classified results of this exercise are contained in [16]. The first four days of this exercise offered an excellent opportunity to test the predictive capability of the SGM. The SGM scenario used for this experiment is summarized in Figure 5-1 and an explanation of the scenario and results follows. (See Volume II, SGM User's Guide, for definitions of SGM scenario parameters.)

Salty Rooster Flying Schedule. We used the SGM to attempt to fly the first four days of the Salty Rooster flying schedule. All weather cancellations were eliminated from the schedule, i.e., any weather cancellation that occurred was treated as if the sortie had never been scheduled.

Hahn AB F-4E Maintenance Data. We used the F-4E maintenance data collected from Hahn AB maintenance tapes, and processed these data through the SGM Maintenance Subsystem to produce aircraft work center break and repair rates. (See Volume V, Maintenance Subsystem for a description of this process.)

- Salty Rooster Flying Schedule, Days 1-4
  Weather Cancellations Elimated
- F-4E Maintenance Data Collected From Hahn AB
  Work Center Break Rates
  Work Center Service Rates
- Salty Rooster Ground Abort Rate
- 20 percent Aircraft Break Rate
- Salty Rooster Initial NMCM Rate
- No Manpower Resource Constraints
- No Spares Resource Constraints

FIGURE 5-1. SGM SCENARIO FOR SALTY ROOSTER TEST

Ground Abort Rate. We used the actual ground abort experienced during the first four days of the exercise.

Aircraft Break Rate. We did not have an F-4E break rate for Hahn AB. However, our visit to the F-4E wing at Seymour Johnson indicated that a typical F-4E break rate ranged from 18 to 22 percent; we used 20 percent.

Initial NMCM Rate. We used the same initial NMCM rate as experienced in the exercise. The appropriate percentage of aircraft were distributed throughout the various work centers at the start of the flying scenario.

Manpower Constraints. We assumed for this experiment that whenever an aircraft broke into a work center there was always a maintenance crew available to begin work on the aircraft, i.e., no aircraft queues ever formed in the work centers. This assumption was made because maintenance manpower was never a limiting factor in sortie production throughout the exercise. This is partially due to the fact that an additional 250 maintenance personnel were brought in for this exercise. In addition, the Hahn AB maintenance

organization was already over 100-percent manned before the augmentation. The majority of the personnel worked 12-hour shifts during the exercise.

Spares Constraints. We assumed that enough spares were available so that no aircraft was NMCS during the first four days. As with the maintenance assumption, no detailed spares data were available and additional spares were brought in for the exercise. The WRSK and BLSS at the base were augmented to the 99-percent level for the exercise; they also brought in a special BLSS of 390 line items consisting of over 13,000 items.

Comparison Results

We compared the SGM predictions using the scenario described versus the actual sorties flown during the first four days of Salty Rooster. Results were compared for each day of the exercise and also for the total sorties flown during the entire four-day exercise. The results of the Salty Rooster exercise are classified Secret; hence, the actual numbers cannot be shown here. However, the SGM results were remarkably close to the actual sorties produced. The cumulative four-day total was within one percent of the actual sorties flown.

Obviously this experiment did not test the accuracy of the model in representing the effects of constraining resources on sortie production. It also didn't test its accuracy in estimating sustainability, since we only looked at four days. However, it did provide a means for judging the major mechanisms in the model: the aircraft maintenance recovery process and the break and ground abort processes. The results of this test provided positive indication that these processes in the model produce reasonable results.

LCOM

The next step in validating the SGM was to run it against the Air Force Logistics Composite (LCOM) Model. LCOM is an extremely powerful and complex

simulation originally developed by the Air Force Logistics Command and RAND during the mid 1960s. This model is used extensively throughout the Air Force for establishing maintenance manpower requirements. LCOM has itself been the subject of considerable validation effort throughout its development and use; hence, LCOM was a natural choice for comparison with the SGM.

To conduct this comparison, we designed a flying scenario jointly with the LCOM group at Headquarters Tactical Air Command (TAC). Then we ran the SGM using this scenario and the notional F-4E data base produced by the SGM Maintenance Subsystem. TAC ran LCOM using the same scenario and the LCOM F-4E surge data base which is a composite of data from four different F-4E bases. Thus, this experiment provided a test of both the model and our notional maintenance data base.

The actual SGM scenario used is shown in Figure 5-2, and the SGM notional F-4E data base is shown in Figure 5-3. Summaries of the common inputs and differences for this experiment are shown in Figures 5-4 and 5-5. Definitions of LCOM inputs are provided in [10]. A discussion of this test scenario and results follows.

Unit Strength (UE). Both the SGM and LCOM used a UE-strength of 46 F-4E aircraft. This is not a standard sized F-4E wing; however, the SGM notional base was developed using this strength to facilitate aggregation of the sortie results from 12 tactical fighter bases. Thus, 46 represents an "average" base in this sense.

Flying Schedule. A 30-day flying schedule of six equally-spaced waves each flying day was used. These waves were spaced 2.5 hours apart, with a constant sortie length of 1.1 hours. The SGM used a minimal recovery period of 0.5 hours. The LCOM sortie leadtime and cancellation time were both 0.5 hours. A sortie leadtime is defined as the amount of time before takeoff when

```
*************************************************************************
***************************   SGM RUN   *********************************
*************************************************************************

SIMULATION -    REPLICATIONS = 100      RANDOM NUMBER SEED = 12.3

AIRCRAFT -    UE = 46    RESERVES =  0    MAXIMUM LAUNCH-SIZE = 46


FLYING SCHEDULE -

         WAVES    TAKEOFF TIMES       MINIMAL     SORTIE   WAIT   OVERNIGHT
DAYS   PER DAY     FIRST    LAST    TURNAROUND   LENGTH   TIME   RECOVERY

 30        6        0630    1630       0.50        1.10    0.40    12.40

RATES -

    INITIAL                     AIRCRAFT
  NMCM RATE     ATTRITION     BREAK RATE    GROUND-ABORT

     0.            0.            0.2000        0.0400


    INFINITE SPARE PARTS ASSUMED FOR THIS SGM RUN.
```

FIGURE 5-2. SGM-LCOM TEST: SGM SCENARIO

work should begin to prepare the aircraft for launch. The cancellation time is the amount of time that will be allowed after scheduled takeoff in order for the aircraft to finish all presortie tasks and takeoff late. The LCOM aircraft were flown in a "clean" configuration, i.e., without external fuel tanks, ECM pods, bombs, etc., to facilitate comparison with the SGM which does not model these items.

Both models were run in the mass-launch mode. The models attempt to fly all mission-capable aircraft for each wave of the day.

Attrition. No attrition was used for this experiment; hence, no reserve aircraft were used either.

NAME	AFSCs	BREAK RATE	REPAIR RATE	AUTH	CREWS	CREWS/ SHIFT
WCS	321X2	0.286	0.1369	34.17	12.47	6
SENSOR	322X2	0.048	0.0948	9.92	4.63	2
AFCS	325X0	0.044	0.1191	8.92	4.05	2
INSTR.	325X1	0.159	0.1379	16.25	7.95	4
ECM	328X3	0.107	0.1516	32.33	12.01	6
COMM	328X0	0.110	0.1402	10.17	5.82	3
NAVIGATION	328X1	0.135	0.1327	17.83	9.07	5
INS	328X4	0.139	0.1437	17.00	8.85	4
PHOTO	404X1	0.011	0.3362	8.83	4.45	2
EGRESS	423X2	0.203	0.1438	22.08	7.63	4
FUELS	423X3	0.059	0.1114	16.33	6.89	3
ELECTRICAL	423X0	0.142	0.1517	12.33	6.00	3
ENVIRONMENTAL	423X1	0.074	0.1507	11.75	5.62	3
PNEUDRAULICS	423X4	0.112	0.1308	18.00	8.41	4
ENGINE	426X2	0.053	0.1030	29.42	10.20	5
MACHINE SHOP	427X0	0.144	0.4619	7.50	4.77	2
STRUC. REP.	427X5	0.172	0.3031	16.08	8.93	4
REPAIR & RECL.	431B1	0.052	0.0826	20.58	7.74	4
A/C FLT	431X1	0.224	0.2196	158.75	92.19	46
WEAPONS	426X0	0.122	0.3625	130.92	34.37	17

FIGURE 5-3. SGM F-4E NOTIONAL BASE

FLYING SCHEDULE

46 UE

30 Flying Days

6 Waves Per Day

0630 First Takeoff Time

1630 Last Takeoff Time

0.5 Minimum Recovery Time (Hours)

1.1 Sortie Length (Hours)

0.0 Attrition Rate

0.0 Initial NMCM Rate

AIRCRAFT MAINTENANCE

Work Center Types

Work Center Authorizations

SPARES

No Spares Constraints For Either Model

FIGURE 5-4. SGM-LCOM TEST: COMMON INPUTS

AIRCRAFT MAINTENANCE (Task Structures, Service Rates, Crew Sizes)

LCOM: F-4E Surge Data Base (3-Base Average)

SGM: Notional F-4E Data Base (12-Base Average)

AIRCRAFT BREAK RATES

LCOM: Failure Clocks

SGM: Post-Sortie = 20 percent; Ground Abort = 4 percent

SIMULATION REPLICATIONS

LCOM: 1

SGM: 100

FIGURE 5-5. SGM-LCOM TEST: INPUT DIFFERENCES

Initial NMCM Rate. Both models assumed all aircraft were fully mission-capable at the start of the scenario.

Aircraft Maintenance. Both models used the same work centers and manpower authorizations, i.e., those for the SGM notional base (shown in Figure 5-3). This decision was made to facilitate comparison between the two models.

The SGM used the work center break rates, service rates, and crews per 12-hour shift shown in Figure 5-3. This notional maintenance data was developed from the maintenance data collected from 12 F-4E bases. (For a description of the construction of an SGM notional base, see Volume V, Maintenance Subsystem.)

LCOM used TAC's F-4E surge data base which contained detailed network description of maintenance tasks for the work centers being modeled. This surge data base was compiled from data from four F-4E bases. LCOM performed only unscheduled maintenance; no phased inspection or scheduled maintenance was performed.

Thus, in summary, LCOM and SGM used the same work centers and manpower authorizations, but different crew sizes, service rates and maintenance task structures.

Failure Rates. The SGM used a post-sortie break rate of 0.20 and a ground abort rate of 0.04, fairly typical rates for the F-4E in a surge environment. LCOM used the failure clock mechanisms inherent in its maintenance task descriptions.

It is unclear what the corresponding LCOM break and ground-abort rates are in that all of this information is implicit in the various LCOM failure clocks and its task network. However, from previous experiments, it

seems that the equivalent LCOM break rate is in the 0.70 to 0.95 range. This is perhaps the most startling difference between LCOM and the SGM.

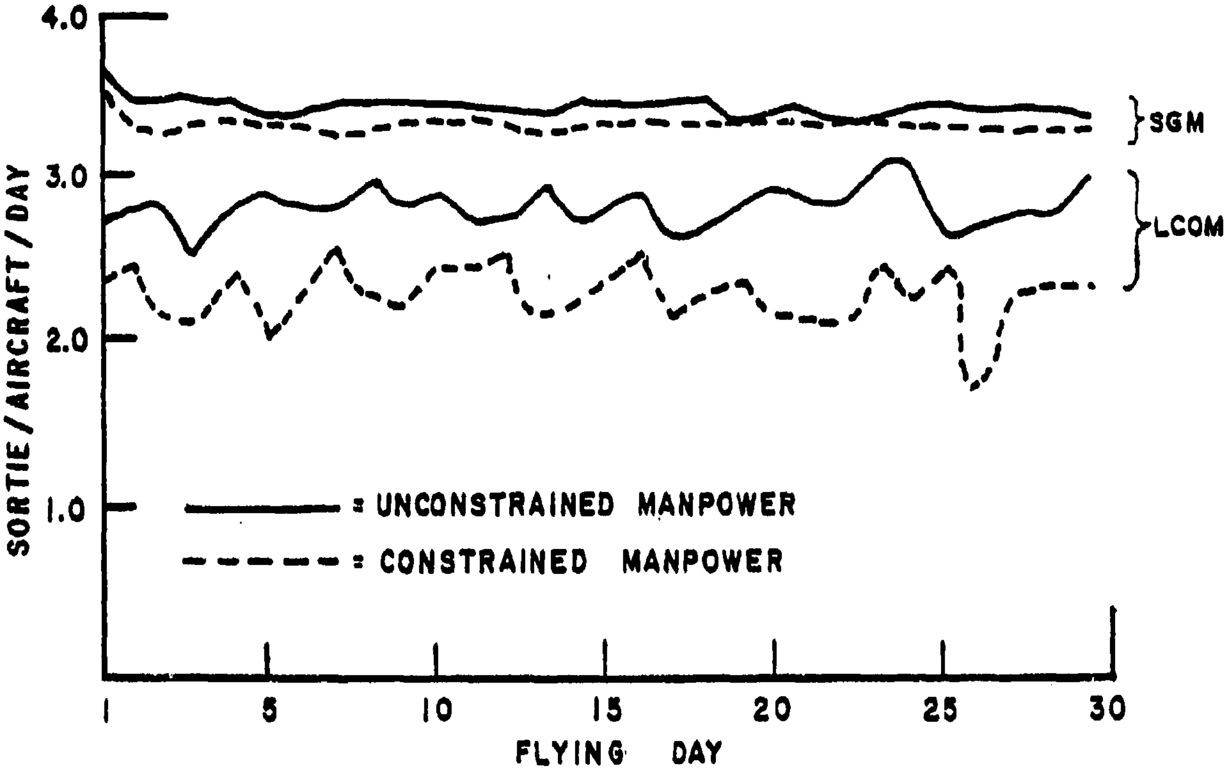
Simulation Replications. The SGM used 100 replications of the experiment and LCOM only one.

Spares Constraints. Both models assumed that there were no spares constraints, i.e., adequate spares were available so that no aircraft was ever NCMS. This assumption was made because no comparable list of spares was available. The SGM models parts at the NSN level while LCOM aggregates parts into higher level Work Unit Codes (WUC).

Comparison of Results. The objectives of this comparison were to test again the major mechanisms of the SGM; in addition, we attempted to test our notional F-4E base for aircraft maintenance. Again, we are not really testing its accuracy in estimating sustainability since no spares were modeled.

Two simulation runs were made with both LCOM and the SGM, one with constrained maintenance manpower and one with unconstrained manpower. By constrained manpower we mean that the work centers are assumed to have the authorized maintenance personnel shown for the SGM F-4E notional base in Figure 5-3. Unconstrained manpower implies that there are enough maintenance personnel in every work center so that no aircraft ever has to wait for a maintenance crew. The results of these runs are shown in Figure 5-6. We have graphed the average number of sorties flown per aircraft for each day of the flying scenario and also computed the 30-day average of this same quantity. As the figure indicates, the SGM produces a much higher sortie rate than LCOM for both the constrained and unconstrained cases.

The obvious questions are what are reasonable sortie rates for this flying scenario and why is the SGM so much higher than LCOM? It would seem



30 DAY AVERAGE SORTIE RATE

	UNCONSTRAINED MANPOWER	CONSTRAINED MANPOWER
LCOM	2.8	2.3
SGM	3.4	3.3

FIGURE 5-6 SGM-LCOM COMPARISON RESULTS

that one would expect a fairly high sortie rate for this scenario. In the unconstrained case, there are no maintenance manpower, spares, aircrew, fuel, or munitions constraints; all aircraft are flying clean, so there are no weapons loading delays. In this highly favorable flying environment, the LCOM sortie rate of 2.8 sorties per aircraft per day seems suspiciously low, especially when compared to the results of previous sortie surge exercises. In fact, when an LCOM comparison was made with those results, the LCOM results were significantly lower than the actual sorties flown.

We feel that the primary cause of the lower LCOM rate is the inherent difference between the LCOM and SGM post-sortie break rates. The SGM used a 0.20 break rate; our visits to the F-4E wing at Seymour Johnson AFB indicated an F-4E break rate ranging between 0.18 to 0.22. Previous LMI experiments with other LCOM data bases indicated that the LCOM rate inherent in the LCOM network and its numerous failure clocks was up in the range 0.7 to 0.95. In fact, when we doubled the SGM break rate, i.e., 0.4 instead of 0.2, we obtained sortie rates of 2.5 and 2.3 for the unconstrained and constrained manpower cases, respectively.

Thus, we feel the LCOM comparison was inconclusive. The LCOM rate was surprisingly low, and we conjecture that it is caused by the inherently higher LCOM post-sortie break rate.

APPENDIX A

RANDOM SAMPLING TECHNIQUES

The SGM requires the generation of random variates according to various probability distributions. The methods used by the SGM were carefully selected according to such criteria as CPU time, memory requirements, and numerical and statistical accuracy. The following paragraphs provide a brief description and appropriate reference for each sampling method.

BINOMIAL

Binomial samples are generated using a combination of two methods. For smaller means, i.e., less than 4.0, a set of Bernoulli samples is generated; for larger means an inverse transform approach is used. These methods are described in [4:418-423].

MULTINOMIAL

Multinomial samples are generated based on a technique known as the Alias method. This method is described in [11:214-218].

NORMAL

A Central Limit approach is used for generating normal variates. Twelve uniformly distributed random variates are summed to produce a normal sample. This method is described in [13:92-93].

POISSON

Poisson samples are generated using a combination of two methods. For Poisson distributions with large means, i.e., greater than 20, samples are generated using a normal distribution with continuity correction as an approximation of the Poisson. For smaller means, a combination of exponentially

distributed samples is used to generate the Poisson sample. These two methods are described in [3:147-156].

UNIFORM (0, 1)

The SGM uses the random number generator supplied from the Honeywell Statistics Library. This generator uses a mixed congruential method and is described in [7:ST194-ST195].

APPENDIX B

BIBLIOGRAPHY

1. Feller, William, An Introduction to Probability Theory and Its Applications, Volume I, New York: John Wiley and Sons, 1968.

2. Fishman, George S., Concepts and Methods in Discrete Event Digital Simulation, New York: John Wiley and Sons, pp. 203-204, p. 224, 1973.

3. __________, "Sampling from the Poisson Distribution on a Computer" in Computing, Vol. 17, pp. 147-156, 1976.

4. __________, "Sampling From the Binomial Distribution on a Computer" in Journal of the American Statistical Association, Vol. 74 No. 366, pp. 418-423, June 1979.

5. Greene, Lieutenant Colonel Richard S., "Aircraft Battle Damage Repair: A Force Multiplier," in Air Force Journal of Logistics, Vol. IV No. 3, pp. 2-8, Summer 1980.

6. Gross, Donald and Carl M. Harris, Fundamentals of Queueing Theory, New York: John Wiley and Sons, 1974.

7. Honeywell, Time-Sharing Applications Library, Volume II - Statistics, DA44, December 1976.

8. Hopkins, LTC Jerry and Major George DeGovanni, "Sortie Surge Analysis (U)," HQ USAF/LEXY/HQ-TAC/XPM (LCOM), SECRET Report, Undated.

9. Johnson, S. E., "The Impact of Battle Damage on A-10 Availability and Sortie Rate," Paper P-1205, Arlington, Virginia: Institute For Defense Analyses, Systems Evaluation Division, May 1976.

10. Keller, Major Kenneth R., Logistics Composite Model Student Training Text, HQ TAC USAF at Langley AFB, Virginia, July 1977.

11. Kronmal, Richard A. and Arthur V. Peterson Jr., "On the Alias Method for Generating Random Variables from a Discrete Distribution" in The American Statistician, Vol. 33 No. 4, pp. 214-218, November 1979.

12. Law, Averill M., "Validation of Simulation Models, I: An Overview and Survey of Real-World Practice", (Contract N00014-76-C-0403, Task NR 047-145), The Office of Naval Research, August 1979.

13. Naylor, Thomas H. et al., Computer Simulation Techniques, New York: John Wiley and Sons, 1966.

14. Read, Robert R., "On the Output of Parallel Exponential Service Channels" in Naval Logistics Quarterly, Vol. 16 No. 4, pp. 555-572, December 1969.

15. Sargent, Robert G., "Validation of Simulation Models" (working paper presented at the Simscript and Simulation Conference), Washington, D.C., March 1978.

16. United States Air Force (Europe), "Salty Rooster Final Report (U)," SECRET Report, August 15, 1978.

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date — Yr. 81, Mo. 09, Day 21

02. Summary prepared by (Name and Phone) — Michael J. Konvalinka (301) 229-1000

03. Summary action — New [X] Replacement [ ] Deletion [ ]

Previous Internal Software ID

04. Software date — Yr. 81, Mo. 09, Day 21

05. Software title — The Sortie-Generation Model System Volume III, Sortie-Generation Model Analyst's Manual

06. Short title

07. Internal Software ID — None

08. Software type
- [X] Automated Data System
- [ ] Computer Program
- [ ] Subroutine/Module

09. Processing mode
- [ ] Interactive
- [ ] Batch
- [X] Combination

10. Application area

General:
- [ ] Computer Systems Support/Utility
- [ ] Scientific/Engineering
- [ ] Bibliographic/Textual
- [ ] Management/Business
- [ ] Process Control
- [X] Other

Specific: Logistics Capability Assessment

11. Submitting organization and address

Logistics Management Institute
4701 Sangamore Road
P.O. Box 9489
Washington, D.C. 20016

12. Technical contact(s) and phone

Mr. John B. Abell
Mr. Michael J. Konvalinka
(301) 229-1000 AV 287-2779

13. Narrative

The Sortie-Generation Model System provides the capability for relating aircraft spares and maintenance manpower levels to the maximal sortie-generation capability of tactical air forces over time.

14. Keywords

Readiness; Resource Allocation; Sortie Generation Capability; Logistics Capability Assessment

15. Computer manuf'r and model	16. Computer operating system	17. Programming language(s)	18. Number of source program statements
Honeywell G-635	GCOS	Cobol 600 Fortran 600/GMAP	15000

19. Computer memory requirements	20. Tape drives	21. Disk/Drum units	22. Terminals
49k words 36 bits each		1 Disk 2 million words	1 time sharing

23. Other operational requirements

24. Software availability — Available [X] Limited [ ] In-house only [ ]

25. Documentation availability — Available [X] Inadequate [ ] In-house only [ ]

26. FOR SUBMITTING ORGANIZATION USE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A111 300	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Sortie-Generation Model System Volume III Sortie-Generation Model Analyst's Manual		5. TYPE OF REPORT & PERIOD COVERED
		6. PERFORMING ORG. REPORT NUMBER LMI Task ML102
7. AUTHOR(s) John B. Abell, Michael J. Konvalinka		8. CONTRACT OR GRANT NUMBER(s) MDA903-81-C-0166
9. PERFORMING ORGANIZATION NAME AND ADDRESS Logistics Management Institute 4701 Sangamore Road P.O. Box 9489 Washington, D.C. 20016		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Assistant Secretary of Defense (Manpower, Reserve Affairs, & Logistics) The Pentagon, Washington, D.C.		12. REPORT DATE September 1981
		13. NUMBER OF PAGES 64
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

"A" Approved for public release

DISTRIBUTION STATEMENT A

Approved for public release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Unlimited

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Readiness; Resource Allocation; Sortie Generation Capability; Logistics Capability Assessment

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Sortie-Generation Model System provides the capability for relating aircraft spares and maintenance manpower levels to the maximal sortie-generation capability of tactical air forces over time.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)